Cyrix III Processor DataBook

Socket 370 Compatible x86 CPU Featuring
MMX™ and 3DNow!™ Technology

Trademark Acknowledgments:

Cyrix is a registered trademark of Via Cyrix Corporation.
Cyrix III is a trademark of Via-Cyrix Corporation. MMX is a trademark of Intel Corporation.
All other brand or product names are trademarks of their respective companies.

Via-Cyrix Corporation
2703 North Central Expressway
Richardson, Texas 75080-2010
United States of America

REVISION HISTORY

Date	Version	Revision
1/25/00	1.0	Final Specs updated for production
3/22/99	0.53	Changed name from MXs to Cyrix III processor.
3/17/99	0.52	Pages 1-9, 1-10: Both L1 and L2 caches are unified. Page 1-11 Remove paragraph concerning Scratch Pad Cache Memory Page 2-36 and similar pages Replace “Don’t care” with xxxxb. Page 2-56 Added Clock Ratio Table for BIOS Core/Bus Frequency Ratios Added CPUPRES# signal. Changed X32 pin to GND Changed Z32 pin to Vcc Changed name VDD-2.5 to VCC_2.5 Changed AD32 pin to Vcc Added IERR# signal Added BRO# Modified Figure 5.3 Voltage Connections Changed REF7 to VREF7 in pin diagrams.
3/16/99	0.51	Corrected Pages 5-1 and 5-2 Pin Assignment Diagrams Corrected Pages 5-3 and 5-4 Pin Signal List Added power diagram Page 5-6 Figure 5-3.
3/11/99	0.5	Page 2-33 Updated and completed CPU Configuration Register Table Page 2-43 Added question marks --Does CCR7 exist? Page 2-48 Updated RCRn bit 0 RCD/RCE Page 2-53 Added Clock Ratio Table for DIR3 TYPE Field
3/9/99	0.4	Bullet Page:Added Programmable Clock/Bus Ratio, new ratio Page 3-1 Made several changes to Figure 3-1. Page 3-2 Removed subname explanation. Page 3-2 Redefined ADS#. Page 3-3 LOCK# is now I/O. Page 3-5 Many changes to Table 3-2, non-supported signals. Page 3-9 Rewrote second paragraph right column. Page 3-10 Omitted table 3-6 as all signals are disconnected during RESET#. Page 3-13 Placed Error Phase before Snoop Phase. Page 3-14 Rewrote INTR explanation. Now there is one interrupt acknowledge bus cycle. Page 3-15 Next to last paragraph. Removed last sentence concerning FERR#. Page 4-2 Made many changes to Table 4-1 Pull-Up Resistors. Page 4-3 Table 4-2 Note 3. Removed the word “APIC”. Page 4-4 Table 4-3, Recommended Operating Conditions for CMOS Signals. Removed V_{CCIO} row. Page 4-5 Moved BSEL signals from GTL I/O to CMOS Input row. Removed LINT signals. Page 4-7 and 4-8 Added 433, 450, 500 MHz frequencies.
3/3/99	0.3	Page 1-9, reworded right-top paragraph concerning exclusive cache. Corrected BSEL0 and BSEL1 typos on page 3-1 and 4-5. Added 133 MHz bus on page 3-9. Corrected note to Figure 4-14 on page 4-12. Updated Figure 4-15 on page 4-11.
3/1/99	0.2	Typos. Added 2.5x to Table 3-3. Updated thermal information.
2/18/99	0.1	Initial Version C:\documentation\joshua\CyrixIII_0.fm

Cyrix III PROCESSOR

Socket 370 Compatible CPU

MMX™ and 3DNow!™ Technology

Cyrix Processors

Introduction

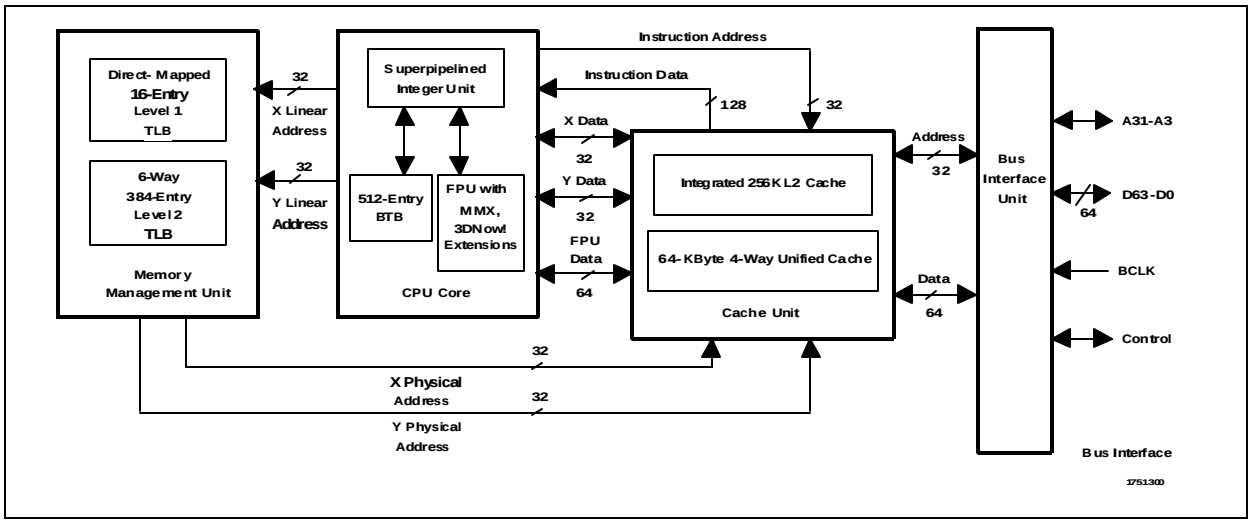
- Performance Features
  - Performance Rating: PR400, PR450, PR500 and higher
  - Integrated 8-way 256 KByte L2 Cache
  - 64K 4-Way Unified Write-Back L1 Cache
  - Branch Prediction with a 512-entry BTB
  - Enhanced Memory Management Unit 2 Level TLB (16 Entry L1, 384 Entry L2)
  - Scratchpad RAM in Unified Cache
  - Optimized pipelining for both 32- and 16-Bit Code
  - High Performance dual pipeline 80-Bit FPU
  - Supports bus speeds of 66, 100, and 133 Mhz

- Other Features
  - Leverages Existing Socket 370 Infrastructure
  - Compatible with MMX™ and 3D Now!™ Technology
  - Runs Windows® 98, Windows 3.x, Windows NT, DOS, UNIX®, OS/2®, and all other x86 operating systems.
  - 2.2 V Core
  - Flexible Core/Bus Clock Ratios: 2.5x, 3x, 3.5x, 4.0x, 4.5x, 5.0x, 5.5x, 6.0x, 6.5x, 7.0x, 7.5x
  - BIOS Programmable Core/Bus Clock Ratio

The Cyrix III processor offers significant performance enhancements over previous generation processors in a Socket 370 compatible package. The Cyrix III includes a 64 KByte L1 cache, an integrated 256 KByte L2 cache, and has frequency scalability to 400 MHz and beyond. It is compatible with MMX and 3DNow! Technology for superior graphics performance. A new dual pipeline FPU/MMX Unit delivers superior floating point performance. The Cyrix III delivers high 32-bit and 16-bit performance while running Windows 98, 95 and 3.X, Windows NT, OS/2, DOS, UNIX, and all other x86 operating systems and applications.

The Cyrix III processor achieves top performance through the use of two optimize superpipelined execution units, two integer units, and a dual issue FPU/MMX/3DNow! unit that can execute instructions from both execution units simultaneously. The 64 K L1 cache and 256 K integrated L2 cache employ write-back technologies to make access to the code and data as fast as possible to avoid pipeline stalls. The cache supports caching SMI code and data, and can be used as scratchpad RAM by the processor.

The superpipelined architecture reduces timing constraints and increases frequency scalability. Advanced architectural technologies include register renaming, out of order completion, data dependency removal, branch prediction, and speculative execution. The pipelining and superscaling are designed to remove data dependencies and resolve conflicts to allow for a high number of instruction executions per clock cycle. This promotes the highest performance for both 32 bit and 16-bit applications.



Cyrix Processors

Table of Contents

Table of Contents

Cyrix Processors

Product Overview

1 ARCHITECTURE OVERVIEW

The Cyrix III processor is a 64-bit, x86 instruction set compatible processor that provides high-performance in a Celeron™ compatible PGA 370 socket. The Cyrix III processor offers an enhanced super-scalar core, and a new pipeline, dual-issue MMX™ and 3DNOW!™ -compatible floating point unit (FPU). The Cyrix III processor can process 57 new multimedia instructions compatible with MMX™ technology.

The processor contains a 64K L1 cache and a 256K L2 cache. It operates at a higher frequency, contains an enlarged cache, a two-level TLB, and an improved branch target cache.

The Cyrix III processor core is an enhanced version of a proven design that offers competitive CPU performance. It has integer and floating point execution units that are based on sixth-generation technology. The integer core contains a dual-issue, seven-stage execution pipeline and offers advanced features such as operand forwarding, branch target buffers, and extensive write buffering. The FPU has been redesigned to provide additional buffering, reduced latency, and improved throughput up to 1 GFLOP (peak). The dual issue FPU can allow two MMX or floating point instructions to execute simultaneously. A 64KB write-back L1 cache is accessed in a unique fashion that eliminates pipeline stalls for fetch operands that hit in the cache.

Through the use of unique architectural features, the Cyrix III processor eliminates many data dependencies and resource conflicts, resulting in optimal performance for both 16-bit and 32-bit x86 software.

To provide support for multimedia operations, the cache can be turned into a scratchpad RAM memory on a line by line basis. The cache area set aside as scratchpad memory acts as a private memory for the CPU and does not participate in cache operations.

The on-chip FPU has been enhanced to process MMX™ and 3DNow! instructions as well as the floating point instructions. Both types of instructions execute in parallel with integer instruction processing. To facilitate FPU operations, the FPU features a 64-bit data interface, a four-deep instruction queue and a six-deep store queue.

For mobile systems and other power sensitive applications, the Cyrix III processor incorporates low power suspend mode, stop clock capability, and system management mode (SMM).

Cyrix Processors

1.1 Processor Differences

Tables 1 describe the major differences between the M II and Cyrix III processors.

Table 1-1. Cyrix III Processor vs. M II Processor

Feature	Cyrix III Processor	M II Processor
Package/pinout	Socket 370 SPGA	Socket 7
Supply voltage	Core voltage = 2.2v i/o reference voltage = 1.0v	Core voltage = 2.9v I/O voltage = 3.3v
CPU primary cache (L1)	64 KB write-back L1 Cache	64 KB write-back Cache
Support for secondary cache (L2)	Internal L2 Cache, 256KBs	Yes (512K external typical)
MMX™ Instruction Set	Yes	Yes
3DNOW!™ Instruction Set	Yes	No
Floating point unit	Dual-issue from Integer Unit	Single-issue from Integer Unit
4MB paging	Yes	No
Virtual Mode Extensions	Yes	No

1.2 Celeron™ Compatibility

The Cyrix III processor is design to be compatible with motherboards created for the Intel® Celeron processor with a socket 370 footprint. However some electrical signaling differs so that the Cyrix III can provide features not supported by the Celeron. In particular, the Cyrix III support two unique pins, the VID[4] pin AK36 used to signal 2.2 volt operation and the BSEL1 pin AK30 used with BSEL0 pin AJ33 to signal system bus frequency.

Conversely, a few minor Celeron signals are not supported by the Cyrix III processor and include: breakpoint signals (BP[3:2] and BPM[1:0]#; internal error signal (IERR#); probe signals (PRDY#, PREQ#); and thermal trip signal (THERMTRIP#).

Refer to chapter 3 of this manual for more details on Cyrix III signal descriptions. For mother-board design considerations and more details concerning Celeron compatibility refer to the *Cyrix III Board Design and AC/DC Specifications* Application Note 120.

1.3 Major Functional Blocks

The Cyrix III processor consists of four major functional blocks, as shown in the overall block diagram on the first page of this manual:

- Memory Management Unit
- CPU Core
- Cache Unit
- Bus Interface Unit

The CPU contains the superpipelined integer unit, the BTB (Branch Target Buffer) unit and the FPU (Floating Point Unit).

The BIU (Bus Interface Unit) provides the interface between the external system board and the processor's internal execution units. During a memory cycle, a memory location is selected through the address lines (A[31-3]#). Data is passed from or to memory through the data lines (D[63-0]#).

Each instruction is read into 256-Byte Instruction Line Cache. The Cache Unit stores the most recently used data and instructions to allow fast access to the information by the Integer Unit and FPU.

The CPU core requests instructions from the Cache Unit. The received integer instructions are decoded by either the X or Y processing pipelines within the superpipelined integer unit. If the instruction is a MMX or FPU instruction it is passed to the floating point unit for processing.

Data is fetched from the 64-KB unified cache as required. If the data is not in the cache it is accessed via the bus interface unit from main memory.

The Memory Management Unit calculates physical addresses including addresses based on paging.

Physical addresses are calculated by the Memory Management Unit and passed to the Cache Unit and the Bus Interface Unit (BIU).

1.4 Integer Unit

The Integer Unit (Figure 1-1) provides parallel instruction execution using two seven-stage integer pipelines. Each of the two pipelines, XandY, can process several instructions simultaneously.

The Integer Unit consists of the following pipeline stages:

- Instruction Fetch (IF)
- Instruction Decode 1 (ID1)
- Instruction Decode 2 (ID2)
- Address Calculation 1 (AC1)
- Address Calculation 2 (AC2)
- Execute (EX)
- Write-Back (WB)

The instruction decode and address calculation functions are both divided into superpipelined stages.

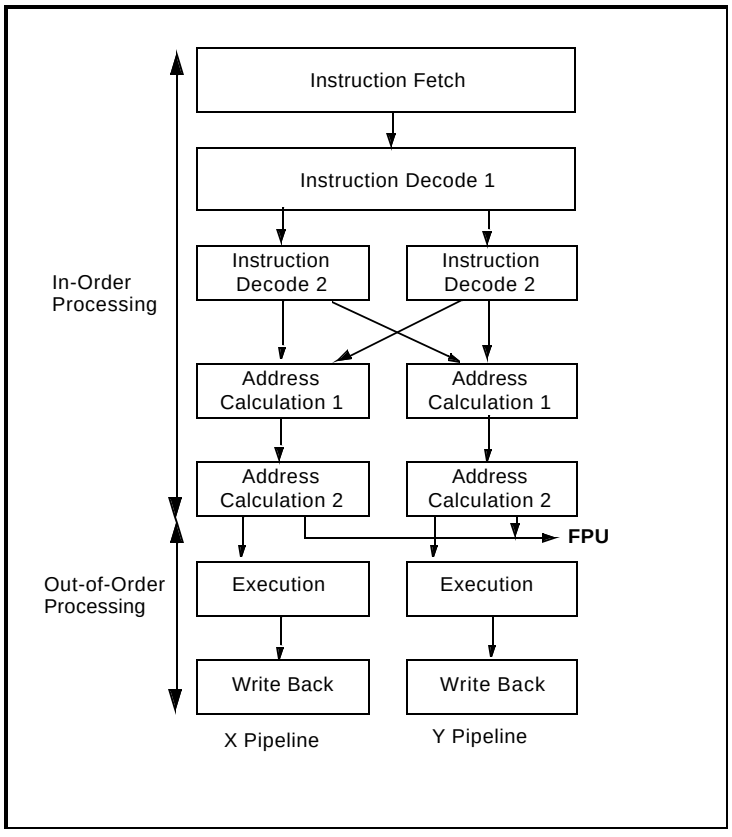


Figure 1-1. Integer Unit

1.4.1 Pipeline Stages

The Instruction Fetch (IF) stage, shared by both the X and Y pipelines, fetches 16 bytes of code from the cache unit in a single clock cycle. Within this section, the code stream is checked for any branch instructions that could affect normal program sequencing.

If an unconditional or conditional branch is detected, branch prediction logic within the IF stage generates a predicted target address for the instruction. The IF stage then begins fetching instructions at the predicted address.

The superpipelined Instruction Decode function contains the ID1 and ID2 stages. ID1, shared by both pipelines, evaluates the code stream provided by the IF stage and determines the number of bytes in each instruction. Up to two instructions per clock are delivered to the ID2 stages, one in each pipeline.

The ID2 stages decode instructions and send the decoded instructions to either the X or Y pipeline for execution. The particular pipeline is chosen, based on which instructions are already in each pipeline and how fast they are expected to flow through the remaining stages.

The Address Calculation function contains two stages, AC1 and AC2. If the instruction refers to a memory operand, the AC1 calculates a linear memory address for the instruction.

The AC2 stage performs any required memory management functions, cache accesses, and register file accesses. If a floating point instruction is detected by AC2, the instruction is sent to the FPU for processing.

The Execute (EX) stage executes instructions using the operands provided by the address calculation stage.

The Write-Back (WB) stage is the last IU stage. The WB stage stores execution results either to a register file within the IU or to a write buffer in the cache control unit.

1.4.2 Out-of-Order Processing

If an instruction executes faster than the previous instruction in the other pipeline, the instructions may complete out of order. All instructions are processed in order, up to the EX stage. While in the EX and WB stages, instructions may be completed out of order.

If there is a data dependency between two instructions, the necessary hardware interlocks are enforced to ensure correct program execution. Even though instructions may complete out of order, exceptions and writes resulting from the instructions are always issued in program order.

1.4.3 Pipeline Selection

In most cases, instructions are processed in either pipeline and without pairing constraints on the instructions. However, certain instructions are processed only in the X pipeline:

- Branch instructions
- Floating point instructions
- Exclusive instructions

Branch and floating point instructions may be paired with a second instruction in the Y pipeline.

Exclusive Instructions cannot be paired with instructions in the Y pipeline. These instructions typically require multiple memory accesses. Although exclusive instructions may not be paired, hardware from both pipelines is used to accelerate instruction completion. Listed below are the Cyrix III CPU exclusive instruction types:

- Protected mode segment loads
- Special register accesses (Control, Debug, and Test Registers)
- String instructions
- Multiply and divide
- I/O port accesses
- Push all (PUSHA) and pop all (POPA)
- Intersegment jumps, calls, and returns

1.4.4 Data Dependency Solutions

When two instructions that are executing in parallel require access to the same data or register, one of the following types of data dependencies may occur:

- Read-After-Write (RAW)
- Write-After-Read (WAR)
- Write-After-Write (WAW)

Data dependencies typically force serialized execution of instructions. However, the Cyrix III CPU implements three mechanisms that allow parallel execution of instructions containing data dependencies:

- Register Renaming
- Data Forwarding
- Data Bypassing

The following sections provide detailed examples of these mechanisms.

1.4.4.1 Register Renaming

The Cyrix III CPU contains 32 physical general purpose registers. Each of the 32 registers in the register file can be temporarily assigned as one of the general purpose registers defined by the x86 architecture (EAX, EBX, ECX, EDX, ESI, EDI, EBP, and ESP). For each register write operation a new physical register is selected to allow previous data to be retained temporarily. Register renaming effectively removes all WAW and WAR dependencies. The programmer does not have to consider register renaming as register renaming is completely transparent to both the operating system and application software.

1.4.4.2 Data Forwarding

Register renaming alone cannot remove RAW dependencies. The Cyrix III CPU uses two types of data forwarding in conjunction with register renaming to eliminate RAW dependencies:

- Operand Forwarding
- Result Forwarding

Operand forwarding takes place when the first in a pair of instructions performs a move from register or memory, and the data that is read by the first instruction is required by the second instruction. The Cyrix III CPU performs the read operation and makes the data read available to both instructions simultaneously.

Result forwarding takes place when the first in a pair of instructions performs an operation (such as an ADD) and the result is required by the second instruction to perform a move to a register or memory. The Cyrix III CPU performs the required operation and stores the results of the operation to the destination of both instructions simultaneously.

Operand forwarding can only occur if the first instruction does not modify its source data. In other words, the instruction is a move type instruction (for example, MOV, POP, LEA). Operand forwarding occurs for both register and memory operands. The size of the first instruction destination and the second instruction source must match.

1.5 Data Bypassing

In addition to register renaming and data forwarding, the Cyrix III CPU implements a third data dependency-resolution technique called data bypassing. Data bypassing reduces the performance penalty of those memory data RAW dependencies that cannot be eliminated by data forwarding.

Data bypassing is implemented when the first in a pair of instructions writes to memory and the second instruction reads the same data from memory. The Cyrix III CPU retains the data from the first instruction and passes it to the second instruction, thereby eliminating a memory read cycle. Data bypassing only occurs for cacheable memory locations.

1.5.1 Branch Control

Branch instructions occur on average every four to six instructions in x86-compatible programs. When the normal sequential flow of a program changes due to a branch instruction, the pipeline stages may stall while waiting for the CPU to calculate, retrieve, and decode the new instruction stream. The Cyrix III CPU minimizes the performance degradation and latency of branch instructions through the use of branch prediction and speculative execution.

1.5.1.1 Branch Prediction

The Cyrix III CPU uses a 512-entry, 4-way set associative Branch Target Buffer (BTB) to store branch target addresses. The Cyrix III CPU has 1024-entry branch history table. During the fetch stage, the instruction stream is checked for the presence of branch instructions. If an unconditional branch instruction is encountered, the Cyrix III CPU accesses the BTB to check for the branch instruction's target address. If the branch instruction's target address is found in the BTB, the Cyrix III CPU begins fetching at the target address specified by the BTB.

In case of conditional branches, the BTB also provides history information to indicate whether the branch is more likely to be taken or not taken. If the conditional branch instruction is found in the BTB, the Cyrix III CPU begins fetching instructions at the predicted target address. If the conditional branch misses in the BTB, the Cyrix III CPU predicts that the branch will not be taken, and instruction fetching continues with the next sequential instruction. The decision to fetch the taken or not taken target address is based on a four-state branch prediction algorithm.

Once fetched, a conditional branch instruction is first decoded and then dispatched to the X pipeline only. The conditional branch instruction proceeds through the X pipeline and is then resolved in either the EX stage or the WB stage. The conditional branch is resolved in the EX stage, if the instruction responsible for setting the condition codes is completed prior to the execution of the branch. If the instruction that sets the condition codes is executed in parallel with the branch, the conditional branch instruction is resolved in the WB stage.

Correctly predicted branch instructions execute in a single core clock. If resolution of a branch indicates that a misprediction has occurred, the Cyrix III CPU flushes the pipeline and starts fetching from the correct target address. The Cyrix III CPU prefetches both the

predicted and the non-predicted path for each conditional branch, thereby eliminating the cache access cycle on a misprediction. If the branch is resolved in the EX stage, the resulting misprediction latency is four cycles. If the branch is resolved in the WB stage, the latency is five cycles.

Since the target address of return (RET) instructions is dynamic rather than static, the Cyrix III CPU caches target addresses for RET instructions in an eight-entry return stack rather than in the BTB. The return address is pushed on the return stack during a CALL instruction and popped during the corresponding RET instruction.

1.5.1.2 Speculative Execution

The Cyrix III CPU is capable of speculative execution following a floating point instruction or predicted branch. Speculative execution allows the pipelines to continuously execute instructions following a branch without stalling the pipelines waiting for branch resolution. The same mechanism is used to execute floating point instructions in parallel with integer instructions.

The Cyrix III CPU is capable of up to four levels of speculation (i.e., combinations of four conditional branches and floating point operations). After generating the fetch address using branch prediction, the CPU checkpoints the machine state (registers, flags, and processor environment), increments the speculation level counter, and begins operating on the predicted instruction stream.

Once the branch instruction is resolved, the CPU decreases the speculation level. For a correctly predicted branch, the status of the checkpointed resources is cleared. For a branch misprediction, the Cyrix III processor generates the correct fetch address and uses the checkpointed values to restore the machine state in a single clock.

In order to maintain compatibility, writes that result from speculatively executed instructions are not permitted to update the cache or external memory until the appropriate branch is resolved. Speculative execution continues until one of the following conditions occurs:

1) A branch or floating point operation is decoded and the speculation level is already at four.

2) An exception or a fault occurs.

3) The write buffers are full.

4) An attempt is made to modify a non-checkpointed resource (i.e., segment registers, system flags).

1.6 Cache Units

The Cyrix III CPU employs two caches, the 64KB L1 Cache and the Exclusive L2 Cache (Figure 1-2, Page 1-10). The main cache is a 4-way set-associative 64-KB unified cache. The unified cache provides a higher hit rate than using equal-sized separate data and instruction caches. While in Cyrix SMM mode both SMM code and data are cacheable.

To avoid data conflicts, both caches are exclusive, that is data can be stored in either cache but not both at the same time.

1.6.1 64-KB L1 Cache

The 64-KB unified write-back cache functions as the primary cache. Configured as a four-way set-associative cache, the cache stores up to 64Kilobytes of code and data in 2048 lines. The cache is dual-ported and allows any two of the following operations to occur in parallel:

- Code fetch
- Data read (X pipe, Y pipeline or FPU)
- Data write (X pipe, Y pipeline or FPU)

The unified cache uses a pseudo-LRU replacement algorithm and can be configured to allocate new lines on read misses only or on read and write misses.

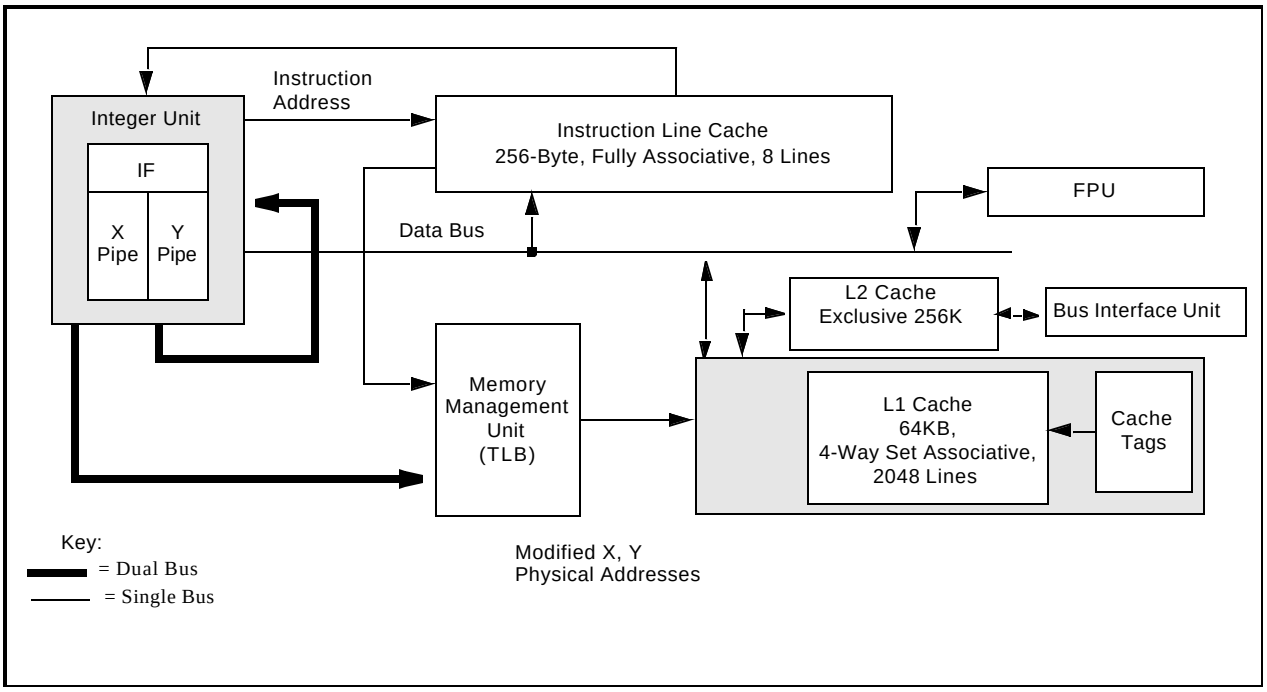


Figure 1-2. Cache Unit Operations

1.6.2 Exclusive L2 Cache

The exclusive 256 KB L2 cache serves as a unified secondary cache. This L2 cache is filled through L1 cache eviction. Fetches from the integer unit that do not hit in the L1 cache will access the L2 cache. The L2 can act as a victim cache, saving cache lines released by the L1 cache. The L2 cache is thus referred to as an exclusive, or victim cache, since it only contains data that is not found in the L1 cache. The L2 cache is 8-way set associative.

The total cache of the Cyrix III CPU can be up to 320k since the exclusive L2 architecture ensures that no data will be in both the L1 and L2. This also eliminates the need for an L1 to L2 writeback cycle, thus improving performance.

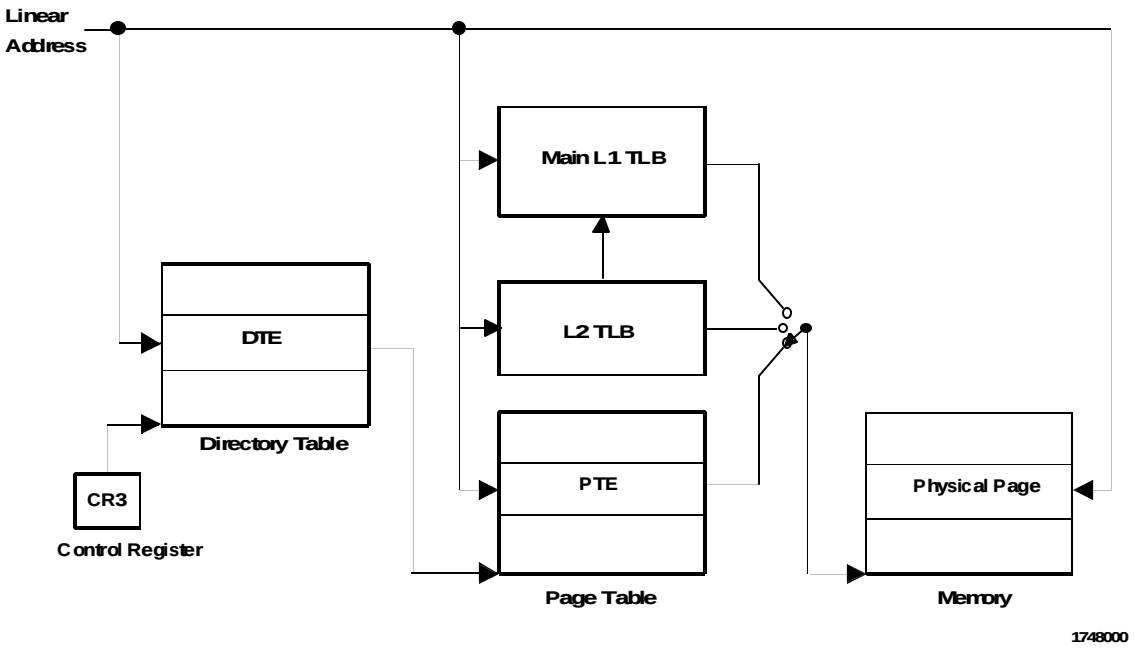
The L2 cache bus operates at the same frequency as the cpu core, delivering cache data at very high speed to the execution units.

1.7 Memory Management Unit

The Memory Management Unit (MMU), translates the linear address supplied by the IU into a physical address to be used by the unified caches and the bus interface. Memory management procedures are x86 compatible, adhering to standard paging mechanisms.

Within the Cyrix III CPU there are two TLBs, the main L1 TLB and the larger L2 TLB. The 16-entry L1 TLB is direct mapped and holds 42 lines. The 384-entry L2 TLB is 6-way associative and hold 384 lines. The DTE is located in memory.

Cache locking is controlled through use of the RDMSR and WRMSR instructions.



1.8 Floating Point Unit

The Floating Point Unit (FPU) processes floating point, MMX, and 3DNOW! instructions. The FPU interfaces to the Integer Unit and the Cache Unit through a 64-bit bus. The FPU is x87 instruction set compatible and adheres to the IEEE-754 standard. Since most applications contain FPU instructions mixed with integer instructions, the FPU achieves high performance by completing integer and FPU operations in parallel.

FPU Parallel Execution

The Cyrix III processor executes integer instructions in parallel with FPU instructions. Integer instructions may complete out of order with respect to the FPU instructions. The Cyrix III processor maintains x86 compatibility by signaling exceptions and issuing write cycles in program order.

FPU instructions can be dispatched from the Integer Unit's X or Y pipeline. The address calculation stage of the pipeline checks for memory management exceptions and accesses memory operands used by the FPU. If no exceptions are detected, the Cyrix III processor checkpoints the state of the CPU and, during AC2, dispatches the floating point instruction to the FPU instruction queue. The Cyrix III processor can then complete any subsequent integer instructions speculatively and out of order relative to the FPU instruction and relative to any potential FPU exceptions which may occur.

As additional FPU instructions enter the pipeline, the Cyrix III processor dispatches up to eight FPU instructions to the FPU instruction queue. The Cyrix III processor continues executing speculatively and out of order, relative to the FPU queue, until the Cyrix III processor encounters one of the conditions that causes speculative execution to halt. As the FPU completes instructions, the speculation level decreases and the checkpointed resources are available for reuse in subsequent operations. The FPU also uses a set of six write buffers to prevent stalls due to speculative writes.

1.9 Bus Interface Unit

The Bus Interface Unit (BIU) provides the signals and timing required by external circuitry. The signal descriptions and bus interface timing information is provided in Chapters 3 and 4 of this manual.